International Journal of Information Technology, Control and Automation (IJITCA) Vol. 6, No.1, January 2016

XML DOCUMENT PROBABILISTIC CLUSTERING BASED ON STRUCTURE AND CONTENT

Hassan Naderi[1] and MojtabaRashidi[2]

[1]University of Science and Technology (IUST), Tehran, iran
[2] Islamic Azad University, Khoramabad, Iran

ABSTRACT

Large volume of information is stored in XML format in the Web, and clustering is a management method for this documents. Most of current methods for clustering XML documents consider only one of these two aspects. In this paper, we propose SCEM (Expectation Maximization Structure and Content) for XML documents which is used to effectively cluster XML documents by combining content and structural features. The other contribution of this paper is that we used probabilistic distributions in such way that have probability parameters corresponding to one cluster. In this way, we obtained better effectiveness compared to other clustering methods due to generality. Experimental results on real datasets show effectiveness of proposed method, particularly when it is applied on large XML documents without schema. Also it can be used to improve accuracy and effectiveness of XML information retrieval.

KEYWORDS

XML, clustering, structural similarity, content similarity, SCEM.

1. INTRODUCTION

Semi-structured nature of XML (extensible Markup Language) documents has converted this language to and standard in presenting and exchanging web information. Wide application of web leads to speed up the research of managing and analyzing XML documents. Hence, mining these documents has become to new scope beside to storing and querying them. XML clustering is grouping the similar data contained in heterogeneous collections without any previous knowledge [1]. XML clustering is useful in different domains such as information retrieval, database indexing, data integration and document engineering [2].

XML clustering is a challenging work compared to Text mining, because these documents have both content information and also structural information. Some methods are presented for XML documents using structural features [4] or content features [5] to separately clustering similar documents. Some research has shown that using only content features don't meet real world application applications. Sometimes, most of the documents are produced only by few schemas. In these situations, XML grouping only based on structural features could lead to incorrect results.

To identify similarity between documents correctly, we should use both structural and content information in clustering process. Methods based on both structural and content features of XML documents have seen very rare [5].

The remainder of this paper is organized as follows. In section 2, we briefly overview some related works about XML clustering. In section 3, we describe content and structure vector model and define similarity measurement for XML documents. In section 4, clustering is done and in

section 5, experimental results are presented. In section 6, we conclude and discuss our future works.

2. RELATED WORK

In recent years, many clustering algorithms are proposed for XML documents, which could be divided in three categories.

Content features based XML clustering: current methods use three approaches for XML clustering using of content features: 1) embedding some special query language such as Xquery in applications. These methods have high cost due to complexities. 2) Mapping XML documents to relation data models. Weakness of these methods is that they ignore semi-structured information contained in XML, which could lead to violating rules in mapping process. 3) Considering XML documents as text and clustering them by traditional text mining techniques. These methods fail to consider semi-structured information of XML documents.

Structure features based XML clustering: These methods mainly focus on two aspects: 1) XML documents presentation. Document layout could be variable and may be modeled by tree, graph, path set, time series, vector and etc. Most of current methods based on tagged tree to present XML documents, because it's a natural presentation and show hierarchical structure of XML document [7]. 2) Measuring similarity and clustering based on structure. First work to clustering structured tree data is designed for XML schema clustering [1]. But it's found that only 48% of documents have relations with special schemas [8]. Hence, integrating large volume of documents without schema and having different semantics to build web database become a tedious work [8]. If solution would be based on tree structure, researches have used tree edit distance to measuring similarity between document structures [7]. Joy Tecly and et al. had worked on similarity measurement for XML documents in [10].

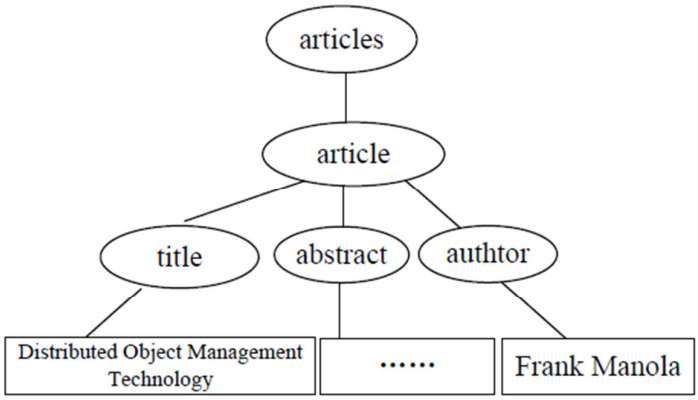
Structural and content features based XML clustering: In spite of advantages in this approach, only few methods have been presented that considered both structural and content features. Reason is that it's major challenge how to effectively combine these two types of features for scalable clustering. Typical methods in this category are: XCFS [2], HCX [11], and SCVM [12].

3. CONTENT AND STRUCTURAL SIMILARITY CALCULATION

We could present XML document as labeled ordered tree like {V,E,R} in which V is nodes set of tag, E is edge sets from parent to child and R is the root of tree. For example, XML document of figure 1 (a) could be presented as figure 1 (b) in the form of a tree [3].

```
<articles>
    <article>
        <title>Distributed Object Management Technology
        </title>
        <author>Frank Manola
        </author>
        <abstract>·····</abstract>
        ·····
    </article>
</articles>
```

(a) An instance of a XML document.



(b) The tree-based presentation of the XML document.

Figure 1: XML document and XML tree.

Given document collection D, each document di could be represent as below:

$$d_i=<v\_struct_i , v\_cont_i> \quad (1)$$

wherev_struct is structure vector and describes document structure, v_cont is content vector and describes document content. These two vectors form content and structure term. Structure term is a path in XML tree from root node to leaf node. For example, structure terms in XML document figure 1 include articles/article/abstract ، articles/article/title ، articles/article/author. Structure space modification is constituted of all structure terms that are extracted from all documents contained in document collection D. We consider structure modification size as 1 and present document structure vector di as below:

$$v\_struct_i =<stw_{i0} ,\dots , stw_{il}> \quad (2)$$

Wherestw_{ij} is the weight of structure modification in d_i.

Term contained in leaf node (that also called text node), is document content term. All terms of all documents contained in document collection D, are extracted and form document content term space. If content term space size is m, content vector of document d_i could be represent as below:

$$v\_cont_i =< ctw_{i0} , ctw_{i1} , \dots , ctw_{im}> \quad (3)$$

where ctw_{ij} is the weight of itm term of content in d_i.

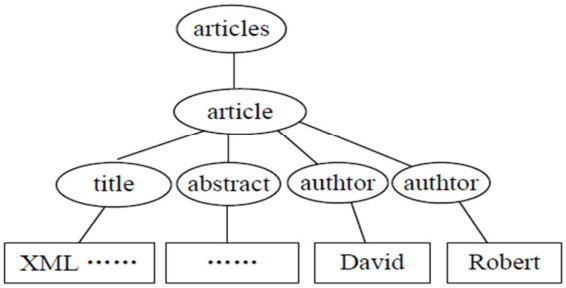
Similarity between XML documents could be present by content vector and structure vector. Because we consider both content and structure information in clustering XML document, accuracy can be improved.

3.1. STRUCTURAL SIMILARITY

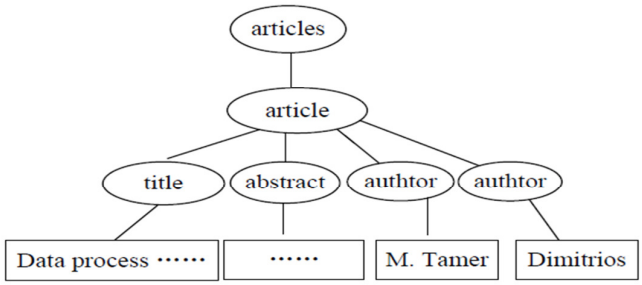
Structural similarity between XML documents could be calculated by term structure vector. Main issue is how to evaluate the weight of each structure term. Observing more frequency in one structure term, in a pair XML documents, does not mean more similarity. For example, even though structure term ‘articles/article/author’ in documents of figures 2a and 2b are seen two time, but it can only say that in document of figure 1, two document of figure 2 have two time

more similarity than doc1 and document of figure 1. In fact, based on content, document of figure 1, is more similar to doc2 (from figure 2b)(both are belonged to data management), hence, only observing or not observing a term in document are considered to evaluating structure term weight.[3] Weight could be defined as below:

$$stw_{ij} = \begin{cases} 1, & if\ st_j\ occures\ in\ d_i, \\ 0, & otherwise \end{cases} \quad (4)$$



(a) the document "doc1"



(b) the document "doc2"

Figure 2: an example of XML document.

Structural similarity between XML documents di and dj is calculated as below by use of cosines size:

$$struct\_sim_{ij} = \frac{v\_struct_i^t \,.\, v\_struct_j}{||v_{struct\,i}|| \,.\, ||v_{struct\,j}||} \quad (5)$$

Where ‖v‖ is normal Euclidean state vector v and v^t is v's transposed.

3.2. CONTENT SIMILARITY

In obtaining content similarity of XML document, content term is related to the current term in text node of XML tree(section 3.1) (including attribute value), hence, content term weights could be evaluated by traditional tf-idf formula [3]:

$$tfidf(ct_j, d_i) = tf(ct_j, d_i).idf(ct_j) \quad (6)$$

Where $tf(ct_j, d_i)$ is content term frequency in document di and idf(ct_j) defined as below:

$$idf(ct_j) = \log \frac{|D|}{df(ct_j)} \quad (7)$$

where |D| is the size of document collection D, df(ctj) is the number of documents that have term ctj. To bound the weight in [0,1] range, we normal it as fallow:

$$idf(ct_j) = \frac{tf(ct_j, d_i).idf(ct_j)}{\sqrt{\sum_{k=1}^{m}(tfidf(ct_k, d_i))^2}} \quad 8$$

Like structural similarity, we could use (5) to evaluate content similarity between documents d_i and d_j.

3.3. XML Document Similarity: Content And Structure Similarity

Based on content and structure similarity definitions, we could evaluate document similarity by putting together these two definitions with special functions. In this paper, we define document similarity as follow:

$$sim(d_i, d_j) = (struct_{sim_{ij}} + cont_{sim_{ij}})/2 \quad 9$$

By use of (9) we obtain content and structure similarity.

4. PROBABILISTIC CLUSTERING

To clustering XML document by SCEM, we need some preprocessing. First, each XML document is divided to content and structural information, then we build content and structure term space. For content information, filtering stop words and stemming are done before term extraction. Terms that occur in lest of the documents or in most of the documents, are removed and then EM algorithm is used to clustering XML documents.
By use of EM algorithm, random values are assigned to ɵ parameters as initial values. Then, M and E steps of this algorithm are continue until parameters would be converged or have very low changes.

In step E, for each data, probability of belonging it to any distribution is calculated as below:[6]

$$p(\theta_j|\theta) = \frac{p(o_i|\theta_j)}{\sum_{l=1}^{k} p(o_i|\theta_l)} \quad 10$$

In step M, parameters are matched to maximizing expected correctness of P(O|ɵ) in above formula. This process is done as below:[13]

$$\mu_j = \frac{1}{k} = \sum_{i=1}^{n} o_i \frac{P(\Theta_j|o_i, \Theta)}{P(\Theta_j|o_l, \Theta)} = \frac{1}{k} \frac{\sum_{i=1}^{n} o_i P(\Theta_j|o_i, \Theta)}{\sum_{i=1}^{n} P(\Theta_j|o_i, \Theta)} \quad 11$$

$$\sigma_j = \sqrt{\frac{\sum_{i=1}^{n} P(\Theta_j|o_i,\Theta)(o_i - u_j)^2}{\sum_{i=1}^{n} P(\Theta_j|o_i,\Theta)}} \quad 12$$

5. CLUSTERING RESULTS AND ANALYZE

In this section, we illustrate the general behavior of the proposed SCEM algorithm. We evaluate our algorithm by using a PC with 2.2 GHz Pentium(R) i5-Core CPU and 4G of memory, running Win7, and programmed by C#.

To evaluate clustering performance, we compare SCEM with three other XML clustering methods. First method only considers structural features by SOMs (self-organizer maps). Second method is traditional content clustering VSM that uses vector space model and tf0idf weight. We compare each algorithm in terms of F1.

Our comparison is based on two real datasets: 1) Wiki10 having 20000 documents into 10 category and 2) XML documents collected by CDISC research group.

To measuring the effectiveness of proposed method, we use F1 measure:

$$F1 = \frac{2 \times Recall \times Precision}{Recall + Precision} \quad (13)$$

Recall equals to ratio between the numbers of correct positive predictions and positive example numbers. And precision equals to ratio between numbers of correct positive predictions and numbers of positive predictions.

Table 1. Clustering result on Texas collection

Dataset	Method	F1
Wiki10	SCEM	0.81
	VSM	0.29
	SOM	0.52
CDISC	SCEM	0.91
	VSM	0.43
	SOM	0.63

To get fairness for all algorithms, we ran each algorithm 10 times on each dataset. Table 1 shows comparison results on real datasets.

Table 1 obviously shows that SOM algorithm is efficient in discriminating structural variations in documents, but unfortunately in case of significant differences in both content and structure of XML document, this efficiency is reduced. Like SOM, VSM that ignores structural information, has very less quality compared to other algorithms. Our proposed algorithm SCEM, uses both content and structural features to improve clustering performance.

6. Conclusion

VSM and SOM are efficient clustering algorithms that are based on either structural information or content information. Unfortunately, due to ignore of content or structure information of XML documents, their accuracy are low. To overcome this problem, we proposed a new clustering algorithm named SCEM. Main contribution of this method is combining content and structural

features and also using of probabilistic technique in clustering XML documents is such a way that each frequent substructure would has a probabilistic parameter for each cluster. Experimental results of real datasets obviously confirm that SCEM is able to cluster XML documents accurately and effectively. Scalability tests also show that this method is scalable and is able to deal with very large datasets. In the case of limited observed data or high number of distributions, the algorithm running would be very costly.

REFERENCES

[1] Aggarwal, C.C, Ta, N, Wang, J, Feng, J, Zaki, M, (2007),Xproj: a framework for projected structural clustering of xml documents. In: Proceeding of the 13th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD 2007, pp. 46–55 (2007).

[2] Kutty, S, Nayak, R, Li, Y, (2009), XCFS - An XML Documents Clustering Approach using both the Structure and the Content. In: Proceeding of the 18th ACM Conference on Information and Knowledge Management, CIKM 2009, pp. 1729–1732 (2009).

[3] Zhang, L, Li,Z, Chen,Q, Li , N, (2010), Structure and content similarity for clustering XML documents, Springer Berlin Heidelberg, 116-124 .

[4] Tran, T, Nayak, R, (2008), Document Clustering using Incremental and Pairwise Approaches. Focused Access to XML Documents. 222-232 (2008).

[5] Doucet, A, Ahonen-Myka, H, (2002), Naive clustering of a large XML document collection. In: Proceedings of the First Workshop of the INitiative for the Evaluation of XML Retrieval, INEX 2002, pp. 81–87 .

[6] NorwatiM.andJalali, M. (2009). Navigation Patterns Mining Approach based on Expectation Maximization Algorithm.

[7] Lesniewska, A, (2009), Clustering XML Documents by Structure. In: Advances in Databases and Information Systems - Associated Workshops and Doctoral Consortium of the 13th East European Conference, ADBIS 2009, pp. 238–246 .

[8] Gan, G, Wu, J, Yang, Z, (2003), The XML web: a first study. In: Proceedings of the 12th International Conference on World Wide Web, WWW 2003, pp. 500–510 (2003)

[9] Hwang, J.H, Ryu, K.H, (2010), A weighted common structure based clustering technique for XML documents. Journal of Systems and Software, 1267–1274 (2010).

[10] Tekli, J, Chbeir, R, Yetongnon, K, (2009), An overview on XML similarity: Background, current trends and future directions. Computer Science Review, 151–173 .

[11] Kutty, S, Nayak, R, Li, Y,(2009), HCX: An Efficient Hybrid Clustering Approach for XML Documents. In: Proceedings of the 2009 ACM Symposium on Document Engineering, DocEng 2009, pp. 94–97

[12] Zhang, L., Li, Z., Chen, Q., Li, N, (2010), Structure and Content Similarity for Clustering XML Documents. In: Shen, H.T., Pei, J., ¨Ozsu, M.T., Zou, L., Lu, J., Ling, T.-W., Yu, G., Zhuang, Y., Shao, J, WAIM 2010. LNCS, Springer,vol. 6185, pp. 116–124.

[13] Han, J,Kamber,M, Pei, J, (2011), Data mining: concepts and techniques: concepts and techniques, Elsevier.